CLAIMS

What is claimed is:

1. A system comprising:

a system memory;

a computer processing module, including:

a host processing element configured to perform a task;

a data-generating processing element configured to perform a subtask within the task, including:

logic configured to receive input data; and

logic configured to process the input data to produce output data, wherein an amount of output data is greater than an amount of input data, a ratio of the amount of input data to the amount of output data defining a decompression ratio,

wherein the output data generated by the data-generating processing element is not contained in the system memory prior to it being generated by the data-generating processing element;

a cache memory coupled to the data-generating processing element for receiving the output data;

a computer processing module interface for outputting the output data from the cache memory;

a communication bus;

a data processing module, including:

a data processing module interface coupling to the computer processing module interface via the communication bus for receiving the output data; and

a data processing engine for receiving and processing the output data from the cache memory, wherein the data processing engine uses a tail pointer to indicate a location within the cache memory from which it has just retrieved data;

wherein, in a write streaming mode of operation, the computer processing module is configured to allocate a portion of the cache memory for the purpose of receiving streaming write output data from the data-generating processing element,

wherein, in the write streaming mode of operation, the system is configured to forward output data from said allocated portion of the cache memory to the data processing module rather than from the system memory, and

wherein the data processing module is configured to forward the tail pointer to a cacheable address of the data-generating processing element, the tail pointer informing the data-generating processing element of the location within the cache memory from which the data processing module has just retrieved data.

2. A system according to claim 1, wherein the host processing element comprises a thread implemented on a computer processing unit, and the data-generating processing element comprises a thread implemented on the same computer processing unit or implemented on another computer processing unit.

3. A system according to claim 1, further comprising plural host processing elements.

4. A system according to claim 3, wherein the plurality host processing elements comprise a plurality of respective threads implemented on at least one computer processing unit.

5. A system according to claim 1, further comprising plural data-generating processing elements.

6. A system according to claim 5, wherein the plurality of data-generating processing elements comprises plural respective threads implemented on at least one computer processing unit.

7. A system according to claim 1, wherein the host processing element and the data-generating processing element each perform functions that are statically allocated.

8. A system according to claim 1, wherein the host processing element and the data-generating processing element each perform functions that are dynamically allocated.

9. A system according to claim 1, further comprising plural data-generating processing elements, wherein each of the plural data-generating processing elements is coupled to the cache memory.

10. A system according to claim 1, wherein the data-generating processing element includes an L1 cache, and said cache memory of the computer processing module is an L2 cache.

11. A system according to claim 10, wherein, in a read streaming mode of operation, the computer processing module is configured to provide the input data by forwarding the input data to the L1 cache of the data-generating processing element, by bypassing the L2 cache.

12. A system according to claim 10, wherein, in the write streaming mode of operation, the computer processing module is configured to forward the output data to the L2 cache by bypassing the L1 cache.

13. A system according to claim 1, wherein the cache memory is an n-way set-associative cache, and wherein the portion is allocated by locking at least one set of the n-way set-associative cache.

14. A system according to claim 1, wherein the allocated portion of the cache memory forms at least one FIFO buffer that couples the data-generating processing element to the data processing module.

15. A system according to claim 14, wherein the system is configured to wrap within said at least one FIFO buffer by using a middle section of an address to index said at least one FIFO buffer, wherein an upper section and a lower section of the address are ignored by the system.

16. A system according to claim 1, wherein the data processing module is configured to process output data received from the cache memory using a modified direct memory access (DMA) protocol.

17. A system according to claim 1, wherein the computer processing module is configured to maintain a cache line state of dirty after accessing a cache line.

18. A system according to claim 1, wherein the decompression ratio is at least 1 to 10.

19. A system according to claim 1, wherein the decompression ratio is at least 1 to 100.

20. A system according to claim 1, wherein the decompression ratio is at least 1 to 1000.

21. A system according to claim 1, wherein the data-generating processing element is configured to dynamically vary the ratio of decompression during its operation in response to at least one criterion.

22. A system according to claim 21, wherein said at least one criterion is depth of scene associated with an object in a scene.

23. A system according to claim 1, wherein the logic for processing the input data further comprises logic configured to execute a dot product operation upon receipt of a dot product instruction using an array of structures computational technique.

24. A system according to claim 1, wherein the logic for processing the input data further comprises logic for compressing data from a first information content amount to a second information content amount to provide the output data, wherein the first information content amount is greater than the second information content amount.

25. A system according to claim 1, wherein the task performed by the host processing element pertains to a graphics processing task, and wherein the subtask performed by the data-generating processing element pertains to the generation of geometry data.

26. A system according to claim 25, wherein the task performed by the host processing element pertains to high level aspects of a three dimensional game application.

27. A system according to claim 25, wherein the logic for processing input data comprises procedural geometry logic configured to transform the input data into the output data, wherein the output data comprises a set of vertices.

28. A system according to claim 25, wherein the logic for processing input data comprises a higher order surface tessellation engine configured to transform information expressed in a higher order surface into output data comprising a set of vertices.

29. A system comprising:

a system memory;

a host processing element configured to perform a task;

a data-generating processing element configured to perform a subtask within the task, including:

logic configured to receive input data; and

logic configured to process the input data to generate output data, wherein an amount of output data is greater than an amount of input data, a ratio of the amount of input data to the amount of output data defining a decompression ratio,

wherein the output data generated by the data-generating processing element is not contained in the system memory prior to it being generated by the data-generating processing element;

a cache memory for storing the output data generated by the data-generating processing element in an allocated portion thereof;

a communication bus;

a data processing engine configured to retrieve the output data from the cache memory via the communication bus, and to process the output data, wherein the data processing engine uses a tail pointer to indicate a location within the cache memory from which it has just retrieved data; and

a tail pointer updating mechanism configured to provide tail pointer updates to a cacheable address of the data-generating processing element via the communication bus.

30. A method for processing data in a system including a host processing element, a data-generating element, and a data processing engine, wherein the host processing element and the data-generating element are coupled to the data processing engine via a communication bus, comprising:

performing a task in a host processing element, the task requiring the execution of a subtask as a part thereof;

performing the subtask in a data-generating processing element when commanded by the host processing element, the performing of the subtask including:

receiving input data; and

processing the input data to produce output data, wherein an amount of output data is greater than an amount of input data, a ratio of the amount of input data to the amount of output data defining a decompression ratio,

wherein the output data generated by the data-generating processing element is not contained in a system memory prior to it being generated by the data-generating processing element;

buffering the output data in an allocated portion of a cache memory;

retrieving, by a data processing engine, the output data from the cache memory via the communication bus, rather than the system memory; and

processing the retrieved output data in the data processing engine, wherein the data processing engine uses a tail pointer to indicate a location within the cache memory from which it has just retrieved data; and

forwarding the tail pointer to a cacheable address of the data-generating processing element, the tail pointer informing the data-generating processing element of

the location in the cache memory from which the data processing engine has just retrieved data.

31. A method according to claim 30, wherein the host processing element comprises a thread implemented on a computer processing unit, and the data-generating processing element comprises a thread implemented on the same computer processing unit or implemented on another computer processing unit.

32. A method according to claim 30, further comprising plural host processing elements.

33. A method according to claim 32, wherein the plurality host processing elements comprise a plurality of respective threads implemented on at least one computer processing unit.

34. A method according to claim 30, further comprising plural data-generating processing elements.

35. A method according to claim 34, wherein the plurality of data-generating processing elements comprises plural respective threads implemented on at least one computer processing unit.

36. A method according to claim 30, wherein the host processing element and the data-generating processing element each perform functions that are statically allocated.

37. A method according to claim 30, wherein the host processing element and the data-generating processing element each perform functions that are dynamically allocated.

38. A method according to claim 30, further comprising plural data-generating processing elements, wherein each of the plural data-generating processing elements is coupled to the cache memory.

39. A method according to claim 30, wherein the data-generating processing element includes an L1 cache, and said above-referenced cache memory is an L2 cache.

40. A method according to claim 39, wherein, in a read streaming mode of operation, the data-generating processing element receives the input data by forwarding the input data to the L1 cache of the data-generating processing element, by bypassing the L2 cache.

41. A method according to claim 39, wherein, in a write streaming mode of operation, the data-generating unit provides the output data by forwarding the output data to the L2 cache by bypassing the L1 cache.

42. A method according to claim 30, wherein the cache memory is an n-way set-associative cache, and wherein the portion is allocated by locking at least one set of the n-way set-associative cache.

43. A method according to claim 30, wherein the allocated portion of the cache memory forms at least one FIFO buffer that couples the data-generating processing element to the data processing engine.

44. A method according to claim 43, further comprising wrapping within said at least one FIFO buffer by using a middle section of an address to index said at least one FIFO buffer, wherein an upper section and a lower section of the address are ignored by the method.

45. A method according to claim 30, wherein the data processing engine processes output data received from the cache memory using a modified direct memory access (DMA) protocol.

46. A method according to claim 30, further comprising maintaining a cache line in a state of dirty after accessing a cache line.

47. A method according to claim 30, wherein the decompression ratio is at least 1 to 10.

48. A method according to claim 30, wherein the decompression ratio is at least 1 to 100.

49. A method according to claim 30, wherein the decompression ratio is at least 1 to 1000.

50. A method according to claim 30, wherein the performing of the subtask includes dynamically varying the ratio of decompression during operation of the data-generating processing element in response to at least one criterion.

51. A method according to claim 50, wherein said at least one criterion is depth of scene associated with an object in a scene.

52. A method according to claim 30, wherein the performing of the subtask involves executing a dot product operation upon receipt of a dot product instruction using an array of structures computational technique.

53. A method according to claim 30, wherein the performing of the subtask involves compressing data from a first information content amount to a second information content amount to provide the output data, wherein the first information content amount is greater than the second information content amount.

54. A method according to claim 30, wherein the task performed by the host processing element pertains to a graphics processing task, and wherein the subtask performed by the data-generating processing element pertains to the generation of geometry data.

55. A method according to claim 54, wherein the task performed by the host processing element pertains to high level aspects of a three dimensional game application.

56. A method according to claim 54, wherein the processing of input data comprises performing procedural geometry to transform the input data into the output data, wherein the output data comprises a set of vertices.

57. A method according to claim 54, wherein the processing of input data comprises performing higher order surface tessellation to transform information expressed in a higher order surface into output data comprising a set of vertices.